

GA Synthesis of Circuits Using a Linear Representation

The procedure for designing many practical circuits can be partly automated.

Ames Research Center, Moffett Field, California

A method of automated synthesis of electronic circuits involves the use of a linear (as defined below) genome representation of circuit elements and of connections among them, plus a relatively simple unfolding technique, in conjunction with a genetic algorithm (GA). The method differs from most other GA-based circuit-synthesis methods in that the topology of a circuit, the number of its components, and the types and values of its components (e.g., inductances, capacitances, and resistances) are all made to evolve by use of the GA.

The linear genome representation is a list of byte codes. The unfolding process is essentially an algorithm in which the byte codes are interpreted to construct a mathematical model of a circuit. In each step of the process, the algorithm starts at one node of the circuit (called the "active node"), proceeds to another node, and connects the two nodes with a component, as instructed by a byte code. The byte code for each component includes (1) an opcode, which specifies the type of component and the identity of the node to which the far end of the component is to be connected, and (2) bytes that specify the value of the component.

The unfolding process begins at a fixed input node and ends at a fixed output node (see Figure 1). In a given step of the process, the far end of a component can be connected to a node that was created previously (e.g., to input, output, or ground), or to a newly created node. When a new node is created, the new node becomes the starting point for the next step. An exception to the unfolding process as described thus far is made for the last component in the list: The last node to be created is connected to the output terminal by a wire. This exception prevents the construction of circuit branches with unconnected ends.

The role of the GA in the synthesis of a circuit is to govern the evolution of both number of byte codes in the list and the specific bytes in each byte code. The GA operates on a population of lists of byte codes, introducing the byte equivalent of mutations. For each member of the population, the unfolding process is carried out, and the electrical performance of the circuit thus synthesized is simulated numerically by the SPICE circuit simulation computer program. The fitness of that member of the population is quantified

by a measure of the difference between the desired circuit output and the actual output according to the simulation. For the sake of speed, the GA is implemented in a master/slave parallel-processing scheme in which a controlling computer generates the members of the population and assigns each member to one of a number of other computers that perform the unfolding process, the simulation, and the evaluation of fitness.

At its present state of development, the method excludes some circuit topologies. Nevertheless, it does enable automated synthesis of many practical circuits that have been designed in the traditional way. For example, one practical circuit synthesized by this method is a low-pass filter for an electronic stethoscope (see Figure 2). Further development of the method can be expected to remove some of the topological restrictions. The incorporation of three-terminal devices (e.g., transistors) has produced amplifier circuits recently.

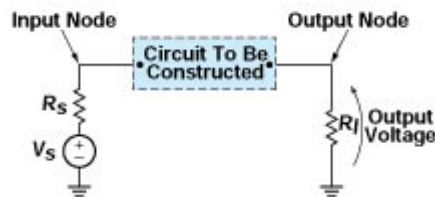


Figure 1. A **Circuit Is Constructed** between fixed input and output terminals. The input source voltage V_s , source resistance R_s , and load resistance R_l are specified in advance.

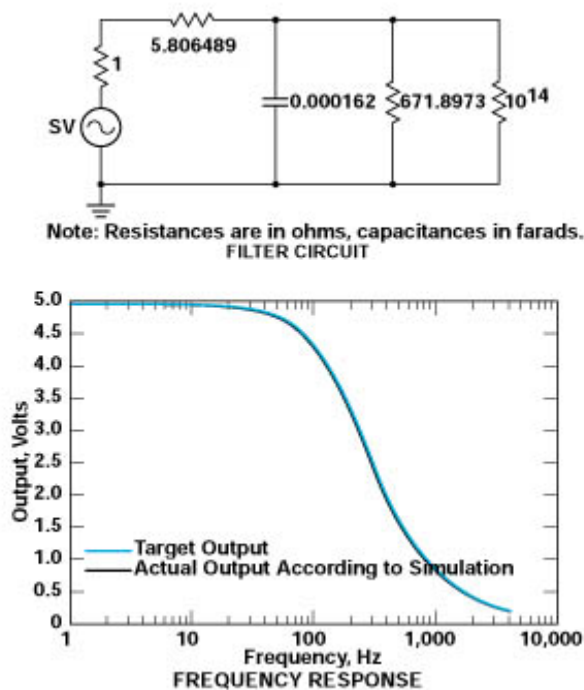


Figure 2. This **Low-Pass Filter** for an electronic stethoscope was synthesized by the method described in the text. The simulated frequency response of the circuit is nearly identical to the target frequency response.